Документация к задаче № 17

Оглавление

[Инструкция по запуску/разворачиванию сервиса у себя на сервере 3](#_Toc169472030)

[Технологический стек решения 5](#_Toc169472031)

[Описание работы и взаимодействия с алгоритма 6](#_Toc169472032)

[Возможный потенциал к развитию 8](#_Toc169472033)

[Данные для авторизации 9](#_Toc169472034)

# Инструкция по запуску/разворачиванию сервиса у себя на сервере

1. Отредактировать файл *.env* под свои нужды.
2. Отредактировать конфигурационные файлы `config.py`,`gunicorn.conf.py`
3. Запустить команду *docker-compose up -d* и дождаться запуска всех контейнеров
4. Подключиться к PosgtreSQL используя данные из файла `.env`, создать базу данных `ship\_tracking` и загрузить в неё дамп `dump-ship\_tracking.dump` из каталога dump (формат custom)

Ввести, отредактировав под свои нужды команду*:   
docker exec -it ui\_ss superset fab create-admin \*

*--username admin \*

*--firstname Admin \*

*--lastname Admin \*

*--email admin@company.ru \*

*--password hfFdmID53CJF; \*

*docker exec -it ui\_ss superset db upgrade; \*

*docker exec -it ui\_ss superset init;*

1. Ссылки для работы в системе:

- ip\_host:9100 - Superset (логин и пароль в пункте 5 текущего документа)

- ip\_host:9101 - Flower (UI)

- ip\_host:9103 **–** PostgreSQL

- ip\_host:9104/docs – API

- ip\_host:9105 - Flower (отслеживание работы Celery)

1. Авторизоваться в системе используя данные из пункта 4, в меню «Настройки» выбрать пункт “Database connections” и добавить базу данных “ship\_tracking” (PostgreSQL)
2. Перейти в раздел “Datasets” и импортировать из каталога dump датасеты из архива dataset\_export\_\*.zip
3. Перейти в раздел “Charts” и импортировать из каталога dump графики из архива chart\_export\_\*.zip
4. Перейти в раздел “Dashboard” и импортировать из каталога dump дашборд из архива dashboard\_export\_\*.zip

# Технологический стек решения

Основной технологический стек решения включает (Рисунок 1):

1. Язык программирования: Python.
2. BI-инструмент: Superset.
3. БД: Postgres (данные по задаче и служебная БД BI-инструмента), Redict (Кэш).
4. Стек библиотек моделей: тumpy, pandas, networknx.
5. Обучение моделей: JupyterLab.

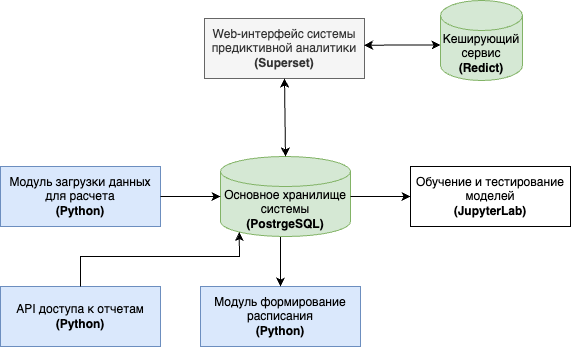


Рисунок 1 Архитектура системы

# Описание работы и взаимодействия с алгоритма

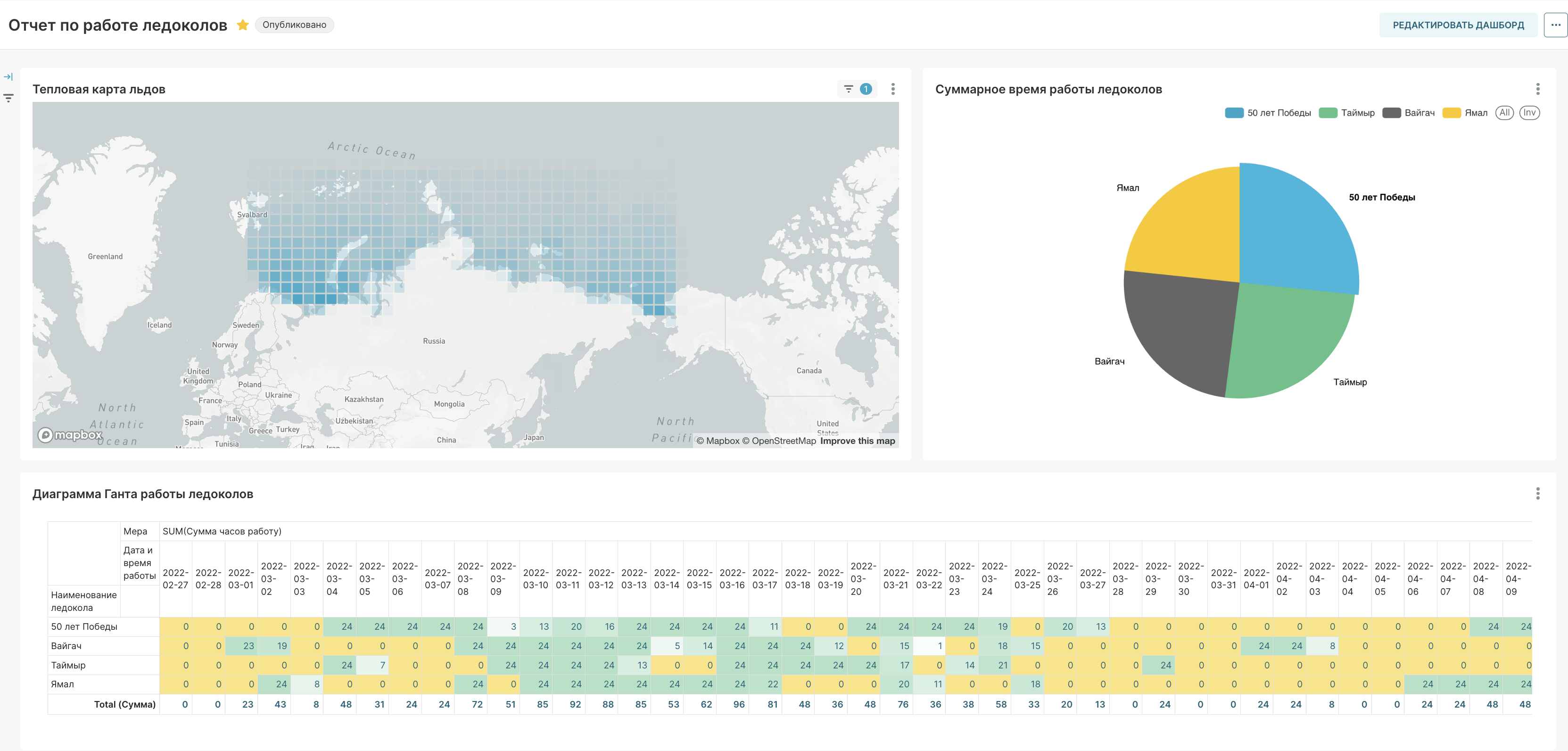
Инструкция по работе с системой представлена по ссылке: <https://disk.yandex.ru/i/qOYQNrGOWyi5bQ>

**Реализованы разные способы проверки алгоритма:**

* В папке notebooks/algorithm/monte\_carlo.ipynb реализован основной функционал алгоритма, который работает с предоставленными файлами.
* Реализован АПИ механизм по работе с расчетами. Спецификацию АПИ можно получить по ссылке: [*https://ai.api.vniizht.ru/docs*](https://ai.api.vniizht.ru/docs) или в *README.md* на гитхабе.
* Результаты работы моделей отображаются в пользовательском интерфейсе: [*https://board.vniizht.ru*](https://board.vniizht.ru)

**Краткое описание пользовательского интерфейса**

* В качестве основного интерфейса выбран *SuperSet*.
* После авторизации в интерфейсе реализован дашборд по ссылке: <https://board.vniizht.ru/superset/dashboard/p/yapVbewkZ6J> (Рисунок 3)



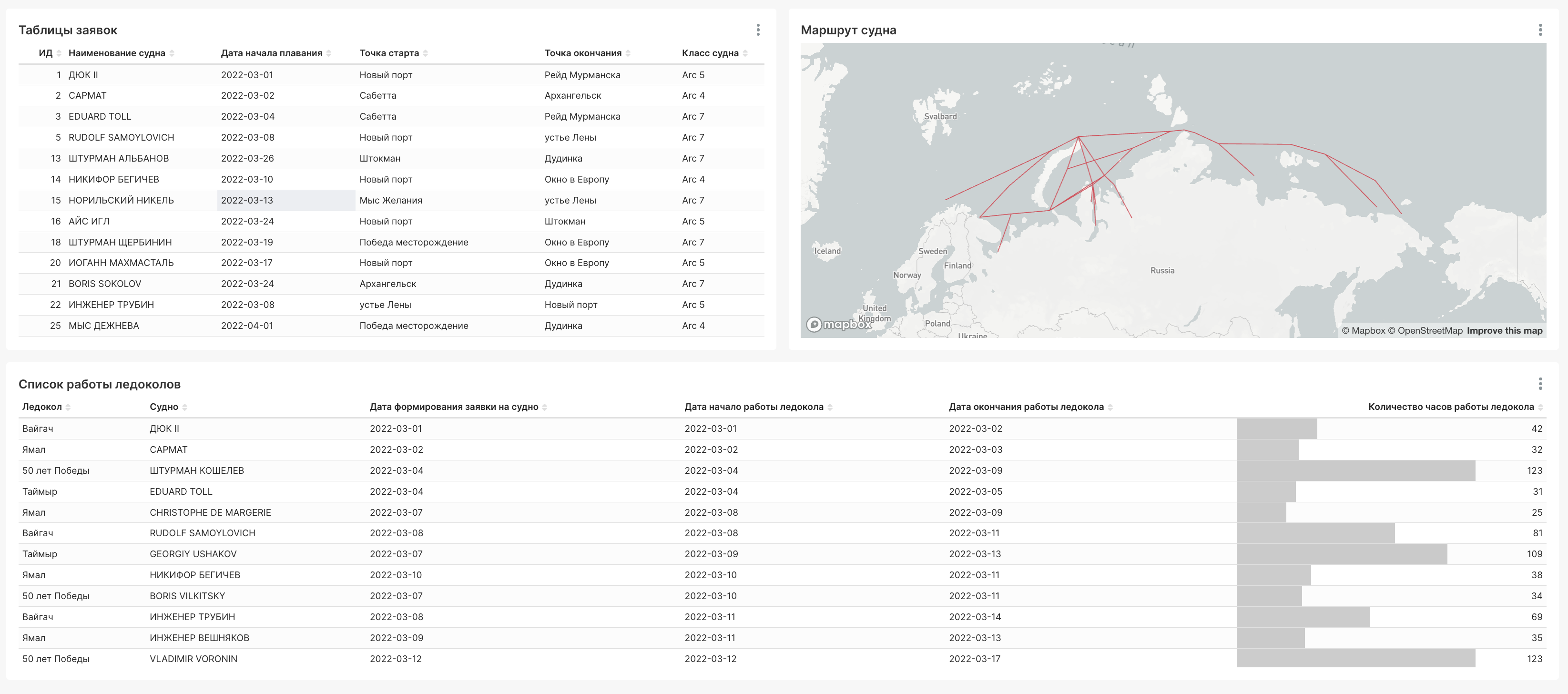


Рисунок 3 Общий вид интерфейса

* В дашборде слева можно найти фильтры по временным интервалам и заявкам по конкретным судам (Рисунок 4).

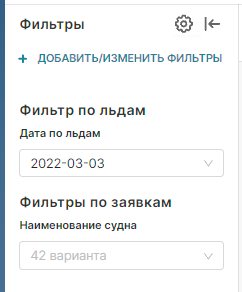


Рисунок 4 Фильтрация в интерфейсе

# Возможный потенциал к развитию

**В части алгоритма**

* Усовершенствование правил игры в части для алгоритма Монте-Карло.
* Автоматизация процесса подбора гиперпараметров алгоритма с использованием кросс-энтропии.
* Ускорение процессов построение графа возможных переходов для судов оборудования.

**В части ПО:**

* Добавление интеграционных механизмов взаимодействия с другими АСУ.
* Автоматизация разворачивания (*CI/CD*).
* Расширение функционала дашбордов.
* Добавление функционал добавления льдов.

# Данные для авторизации

1. Web-интерфейс:

- Логин: rosatom

- Пароль: 123456Qw@

2. АПИ по получению расчетов:

- Логин: present

- Password: @Z&WSSv?5|AVhTeD